NISTIR 4308

THE CELL AS PART OF A MANUFACTURING SYSTEM

Albert Jones

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Center for Manufacturing Engineering
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Dr. John W. Lyons, Director

NIST

NISTIR 4308

THE CELL AS PART OF A MANUFACTURING SYSTEM

Albert Jones

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Center for Manufacturing Engineering
Gaithersburg, MD 20899

May 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Dr. John W. Lyons, Director

THE CELL AS PART OF A MANUFACTURING SYSTEM

Albert Jones
National Institute of Standards and Technology
Gaithersburg, MD 20899
USA

Abdol Saleh
AT&T Bell Laboratories
Holmdel, NJ 07733
USA

1.0 INTRODUCTION

In this paper, we describe a new approach to the design, implementation, and integration of cell controllers in a manufacturing system. It combines techniques from control theory, operations research, and computer science. This cell controller can be 1) modified to fit both the physical definition of a cell and the capabilities of other controllers in the system, and 2) easily integrated into any shop floor control system which meets the interface requirements.

1.1 The Automated Manufacturing Research Facility

The results described in this paper are based on the experience gained in building the Automated Manufacturing Research Facility (AMRF) [SIM82] at the National Institute of Standards and Technology in the US. The AMRF is a prototype small batch manufacturing facility built to address two important issues: integration standards and measurement techniques in an automated factory. Physically, the AMRF contains a variety of robots, machining centers, a coordinate measuring machine, and an automated guided vehicle. This equipment has been integrated together using three separate architectures: shop floor control, data management, and communications (see Figure 1).

The AMRF shop floor control architecture is a four level hierarchical system [JON86] (see Figure 2). Currently, only the bottom three levels have been implemented. Each piece of equipment has its own AMRF-built controller. This equipment is grouped into small units called workstations. Each workstation is designed to perform a specific activity (eg milling, turning, inspection, material transfers etc.) and has its own controller. The workstations are managed by the cell controller. All planning and scheduling decisions are made at cell level. Each workstation is sent one instruction at a time. It first decomposes that instruction into the tasks

to be performed by the equipment under its control. It then coordinates the activities of those equipment as required. The task decomposition is, in general, completely deterministic for each set of task related-data with little or no flexibility.

AMRF researchers have designed and implemented an architecture called IMDAS - the Integrated Manufacturing Data Administration System - to manage all data [LIB88]. IMDAS has been specifically designed to operate in a distributed, heterogeneous computing environment in which 1) control computers have time critical data needs, and 2) data resides in a variety of commercial databases. IMDAS is completely separate from the control hierarchy and transparent to the modules in that hierarchy (its users). Users simply request data from IMDAS in a standard way. IMDAS then retrieves the data wherever it is and provides it to users in the format they desire. Users are totally unaware of the effort required to answer their requests. The AMRF view is that IMDAS plays the same role in managing its resources (data and data repositories) that the shop floor control hierarchy plays in managing its resources (material & equipment). Hence, IMDAS is a three level hierarchy of data management services: the Basic (BDAS), the Distributed (DDAS), and the Master (MDAS) Data Administration Service modules (see Figure 3). Each BDAS can be tied to multiple data repositories. Detailed descriptions of these functions can be found in [LIB88].

In the AMRF, processes communicate with each other by writing and reading messages in memory areas that are accessible by both the process and the communications system. These "common" memory areas are called mailboxes. The network communications system is responsible for delivering messages from the source mailboxes written by applications processes to any destination mailboxes that are logically connected to them. For those processes residing on the same computer system, this is very simple. For those processes residing on different computer systems, an external network is required. The AMRF network architecture contains several subnetworks linked to a large backbone network (see Figure 4). Each workstation has its own local area network linking its equipment controllers. These subnets are either RS232 or Ethernet and ensure quick response for time critical operations. The backbone network is a broadband, token-bus network. Details about the evolution of the AMRF network can be found in [RYB88].

We will return to the AMRF later in this paper.

1.2 Overview

In section 2 we describe a cell. In section 3, we examine the classical approaches to controlling dynamic systems and review some applications, to-date, in manufacturing. Section 4 contains a description of the two most popular approaches to shop floor control: hierarchical and heterarchical. In section 5 we detail our approach to designing a cell controller and discuss information requirements and implementation issues. We also provide a summary and bibliography.

2. WHAT IS A CELL

Currently, there is no standard, or even accepted, definition of a cell. In industry, one finds two types of cells. They contain either a collection of identical, or functionally identical, equipment or they are group technology cells. In the first case, a cell might contain a collection of the drills, or a collection of milling machines, etc. In the second case, a cell will contain all of the equipment (sometimes a single piece) needed to manufacture a particular family of parts, assemble a specific family of circuit boards, etc.

The AMRF introduced a totally different kind of cell. The AMRF cell contains physical groupings of equipment called workstations (see Figure 5). There are four types: machining, cleaning/deburring, inspection, and material storage/transportation. The architecture described below can be used for each of these "cells".

There are many companies marketing "cell" controllers even though there is no standard definition of a "cell". In addition, there are no standard internal functions, external interfaces, or hardware platforms. There are two classes of vendors, and two different design philosophies. Vendors of equipment controllers have based their cell controller designs on their existing Programmable Logic Controllers (PLC). Those extensions provide the communication and database access necessary to interface with and coordinate several lower level PLCs. Many of them even provide limited scheduling. The controllers marketed by "system integrators" typically contain sophisticated scheduling and database capabilities. Some even use expert systems.

The major problem with this scenario, at least from the users perspective, is that it is virtually impossible to determine what type of cell controller to buy and how to integrate it into an existing or future manufacturing system. The architecture presented in this paper also addresses this issue.

Before we describe our cell controller, we discuss two important background topics: decentralized control of dynamic systems and shop floor control systems.

3. DECENTRALIZED CONTROL of DYNAMIC SYSTEMS

Sandell et al [SAN78] distinguish between two types of decentralized methodologies to control the evolution of dynamic systems: multi-layer and multi-level. Multi-layer controllers deal with the fact that decisions are made and events occur at different frequencies in the same system. However, these types of controllers typically do not specify how decisions are related to one another or how events influence those decisions. Multi-level control, on the other hand, provides a methodology for decomposing complex decisions into smaller, simpler ones, and, in certain cases, solving them to optimality. There is, in general, no dependence on time or frequency. But, as we will see, it can be used to model the relationships between different

decisions at the same frequency and the same decision at different frequencies.

3.1 Multi-layer control

Important events which influence the behavior of large, complex, interconnected systems typically occur at different time scales. Modeling these systems often begins by defining state variables and state transition functions for these events. Events which occur at the same, or nearly the same frequency, can be clustered into groups. These groups form the layers of a multi-layer controller. Each layer operates on a different "time-scale" and uses different sets of aggregated information. The assumption is that state variables within a particular layer have "strong" interactions and those across layers have only "weak" interactions. More details on the mathematical models and structure of these controllers can be found in [JAM83].

Albus [ALB81a] and Saridis [SAR85] pioneered the use of multi-layer control in robotics. They both used a three layer model. In the Albus model those layers are called TASK, E-MOVE, and PRIMITIVE. The TASK layer determines a plan, a series of robot moves, for executing each new robot command. The E-MOVE layer determines an optimal trajectory for each of these moves. The PRIMITIVE layer provides the interface to the robot and monitors the execution of each move. In the Saridis model the three layers are called ORGANIZATION, COORDINATION, and EXECUTION. They perform very similar functions to those found in the Albus model. The ORGANIZATION layer does planning. The COORDINATION layer chooses actions to carry out the plan, and the EXECUTION layer interfaces to a specific robot.

Gershwin [GER89] recently used the notion of multiple time-scales to propose a mathematical justification for hierarchical analysis of production systems. The formulation of the decisions in this system contain both continuous and discrete variables. Furthermore, those decisions can contain deterministic, stochastic, linear, or non-linear terms. Gershwin used the frequency separation methodology discussed above to propose his hierarchy. He placed events that occur very infrequently at higher layers and those that happen very frequently at lower levels. The mathematical relationships needed to control events at higher layers ignored the details of the variations of the events occurring at the lower layers. The formulations at the lower layers viewed the events at the higher layers as static, discrete events.

Villa and Rossetta [VIL86] addressed the temporal relationships that exist between the layers inside a multi-layer controller. They proposed that each controller have three planning parameters: a planning horizon H, an updating period P, and a sampling period, T. The authors argued that a controller will perform efficiently if H>10*P and P>10*T. They also indicated that one could have more than one multi-layer controller superimposed on top of one another. They proposed that the interfacing between controllers in adjacent levels could be achieved by setting a lower level's planning horizon and updating period to the upper level's updating and sampling periods, respectively.

3.2 **Multi-level control**

Mesarovic et al [MES70] presented one of the earliest, formal, quantitative treatments of multi-level control systems. The techniques for problem decomposition are based on methods from the theory of decomposition for mathematical programming problems [GEF70]. The purpose is to decompose a complex optimization problem into a series of smaller and simpler sub-problems. Most decomposition procedures result in a two level structure, (see Figure 6) and use conditioning of either the objective function [DAN60] or the constraint set [BEN60].

Hax and Meal [HAX75] were one of the first to apply these concepts to a production planning problem. It is important to note that their decomposition of the production planning problem led to a similar aggregation/disaggregation (tree structure) of the information about the end products to be produced. A product was first classified by type. Each type has one or more "families". Each family has one or more "items". At each level, a mathematical programming problem is formulated to solve the resulting planning problem. The solution at one level provided constraints to the next lower level. Several authors, including [BIT77 and AXA81], have discussed conditions under which decompositions and aggregations of this kind guarantee that solutions will exist at each level.

Davis and Jones [DAV88] used this approach to decompose scheduling into a two-level decision-making problem (see Figure 6). They use both price-directed and goal-directed methods to ensure coordination of objective functions and constraints. They have also discussed the impact of this on the structure and content of the process plans that are used [JON89] in the scheduling. The top level in this decomposition, the inter-process coordinator, uses coupling constraints to generate limit times - earliest start and latest finish times - for the tasks assigned to the lower levels. Each process module in the lower level solves a sequencing problem using those limit times as additional constraints. Real-time simulation is used to predict the impact of using various scheduling and sequencing rules on each level in the system. An important feature of this work is the possibility of expanding this approach to more than two levels.

4. SHOP FLOOR CONTROL STRATEGIES

As noted above, cell controllers must fit into some type of shop floor control system. Two approaches have been proposed: hierarchical and heterarchical.

4.1 **Hierarchical Control**

Almost all of the proposed shop floor control architectures use both multi-layer and multi-level control to form hierarchical control systems. They have a tree structure corresponding to the specific arrangement of equipment in the shop. In addition, the designs seem to be based on three guidelines [ALB81b]: 1) levels are introduced to reduce complexity and limit responsibility, decision-making, and authority; 2) each level has a distinct

planning horizon which decreases as you go down the hierarchy; and 3) control resides at the lowest possible level. The application of these guidelines has led to a variety of different architectures. Major differences exist in the number of levels and functions assigned to each level, control paths between supervisors and subordinates, and their handling of data and communications. At the moment, there are no quantitative methods available to compare different designs or to determine the "best" design for a particular application.

Most implementations of hierarchical control principles have two major limitations. First, all decisions are made at the top level. Controllers in lower levels simply execute one command at a time. Furthermore, these controllers have very limited ability or authority to react to the dynamic evolution of the environment in which they operate. Second, the only exchange of data allowed in this type of system is between a supervisor and its subordinates. This means that a supervisor must transmit all data needed to execute a command along with the command. This lack of peer-to-peer communication means that subordinates are cut off from their chain of command whenever the communication link goes down. In addition, since most controllers can only execute one command at a time, part or all of the system can deadlock very quickly.

4.2 Heterarchical Control

Recently, some researchers [HAT85, DUF86] have attempted to address these problems. In addition, they seek to eliminate the rigid supervisor/subordinate relationships found in hierarchies. To do this, they have developed theoretical foundations for and advocate the use of heterarchical structures. The principal characteristics of this type of structure is that all entities are treated as co-operating equals. Decisions regarding what to manufacture, how to manufacture, and when to manufacture are made by committee. The committee carries out an extensive and complicated negotiation process to arrive at those decisions. This approach results in "arbitrary control paths" which, researchers claim, can overcome the potential for system deadlock that exists in most implementations of hierarchical control. Although researchers have demonstrated this approach in the laboratory, they have not shown it to be practical in a real manufacturing environment.

4.3 The AMRF approach

As noted above, the AMRF has implemented a multi-level shop floor control hierarchy (see Figure 2). The AMRF has addressed both of the limitations described above, not by abandoning the hierarchical control approach, but by implementing advanced technologies.

As noted above, the cell controller (level 3) in the AMRF control hierarchy is the only controller that makes any real time decisions - it does scheduling. All other decisions about how and when to perform a particular activity are made off-line. Recently, AMRF researchers have begun to integrate distributed decision-making into the existing hierarchy. Each controller will eventually choose, from alternatives given in a process

plan, HOW to complete assigned jobs. It will then use that plan, together with start and finish times from its supervisor, to determine an exact schedule. The framework outlined in [DAV88] will be used to distribute both of these decisions across the AMRF shop floor hierarchy.

As for the peer-to-peer communication problem, this came about because of hard-wired communication links. Since communication could occur only between two entities that were physically wired together, it was necessary to transmit all data needed to do a job. For example, suppose a cell wanted to send a command to a machining center to machine a part. The cell had to tell the machine tool controller what to do, when to do it, and how to do it. That is, the control path and the data path were the same physical wire. The AMRF researchers recognized early that the main difficulty in implementing separate control and data paths was not hierarchical control, it was technology. They recognized that more sophisticated computing and communication technologies would be forthcoming. So they designed separate architectures for data management, shop floor control, and network communication [BAR89]. This, in effect, allows the control paths to be hierarchical and the data flow paths to completely arbitrary. This means that 1) information can be exchanged between modules anywhere in the AMRF and 2) control interchanges can be restricted to a supervisor and its subordinates.

5. A NEW CELL CONTROL ARCHITECTURE

In this section, we describe the cell controller as one module inside a multi-level, hierarchical, shop floor control architecture. We also include a list of additional assumptions and discuss the influence of the AMRF on this architecture. We conclude with a description of the external interfaces and internal implementation structure for the cell.

5.1 Major assumptions

Each cell controller must be viewed as one part of a larger shop floor control architecture. We assume that architecture has a multi-level structure like the one shown in Figure 2. As shown, each module is simultaneously a supervisor to many subordinates and subordinate to one supervisor. Each module tries to make optimal use of subordinate resources to complete jobs assigned by its supervisor. As described in [JON90], each module will eventually perform three functions:

planning: generate and update a plan for executing assigned jobs
scheduling: evaluate proposed plans, generate/update schedules
regulation: interface with subordinates, monitor progress

These functions generalize those developed in [ALB81, SAR85]. In most applications, they will be performed at different frequencies - regulation the most frequent and planning the least frequent. Hence, we can treat each function as a separate layer in a multi-layer controller. The frequency with which they are actually executed is implementation dependent.

We assume, for the sake of this discussion, that cell controllers form

the second level in the shop floor control system. We stress, however, that this approach is compatible with any of the definitions given above. Each cell has a fixed set of subordinates, the equipment level controllers. We assume that both the cell and equipment controllers execute the three functions described above. If equipment controllers cannot perform these functions (which is the case in many systems today) then the cell must take on this responsibility as well. The cell's supervisor can also be a multi-layer controller, but this is not necessary. We need only assume that the cell receives a list of jobs to do with due dates and priorities.

5.2 Cell Functions

5.2.1 **Planning.** The Planning function (PF) determines a "production plan" for each job assigned by the cell supervisor and updates an existing production plan to account for unexpected problems with subordinate equipment. (see Figure 7). A production plan contains

- o a list of tasks which must be executed to complete the assigned job
- o task assignments for each piece of equipment
- o any precedence constraints among the tasks
- o proposed start and finish times for each task

The tasks in a given production plan become the jobs for the equipment controllers. They in turn will plan, schedule, and execute all of the operations necessary to complete each task.

For a new job, this involves several steps. First, the PF retrieves or generates one or more candidate production plans. In the near future these candidates will be contained in the process plan (see below) for the job. Later, the PF may have the intelligence to generate these candidates in real-time. These candidates are passed to the Scheduling function (SF) which estimates their impact on the evolution of the system. The estimate is based on one or more performance criteria specified by the PF. The selected plan is put into the database for later use in constructing the run-time schedule.

Information is provided by the SF on the status of all jobs and all subordinates. Whenever a problem occurs that cannot be resolved by the SF, the PF must determine a new course of action. It may change job priorities, performance measures, and/or existing production plans. Whenever shop floor conditions are such that the PF can devise no strategy which does not violate one or more due dates, the cell's supervisor must be informed. They will either negotiate a new set of due dates, a new set of jobs, or both.

5.2.2 **Scheduling function.** The Scheduling function (SF) performs three major functions (see Figure 8). It evaluates proposed production plans from the planning functions. It generates a schedule containing a list of tasks with start and finish times for each equipment controller. Finally, it tries to resolve any conflicts and problems with the current schedule identified by the Regulation function (RF).

As discussed above, the SF evaluates candidate production plans for

each job. We expect this evaluation to be carried out using a simulation analysis for a specified period into the future. This analysis is performed to determine the impact of a particular plan on the forecasted evolution of the system during that period, ie the schedule. The PF provides the performance measures and candidate plans to be used in the evaluation, and its best guess regarding the up/down time of all equipment during the analysis period. Performance measures can include tardiness of current jobs, utilization and capacity of equipment, load on the system, and throughput, among others. The SF will prioritize candidate plans based on the selected performance measures. Once the PF selects the production plan to be used it must notify the cell's supervisor of the expected completion time for that job.

Before tasks in a particular production plan can be released to equipment controllers, the SF must compute the anticipated start and finish times of those tasks. That is it must update its current schedule using the performance measures and scheduling rules provided by the PF. These times will be used by the equipment controllers in determining their own plans and schedules. These times are also passed up to the PF as part of the feedback information.

Manufacturing equipment are subject to random failures which cause delays. These delays, if they are long, can make the current schedule infeasible. The SF must resolve these infeasibilities as quickly as possible. A two step process is envisioned. First, the number of times in the current schedule which will be impacted by this delay must be determined. The outcome of this analysis determines the next step. In some cases there may be enough slack in the original schedule to absorb the ripple effect of the delay. In other cases, a new schedule can be generated [DAV88] by simply selecting a new rule from the existing candidate list. Whenever this can not be done, PF and SF may negotiate new start and finish times, the PF or SF may change the existing performance measures, or the PF must specify a new production plan.

5.2.3 **Regulation function.** The Regulation function (RF) is the interface between the cell controller and its subordinate equipment controllers (see Figure 9). It releases jobs to subordinates: monitors subordinate feedback on those jobs; and informs the SF of any problems. The job release strategy depends on the capabilities of the subordinate. If the subordinate can only manage one job at a time, which is the case with most equipment controllers today, then the RF will release one job at a time. A new job is released when the previous one is completed. Feedback data from subordinates is compared with the current schedule to determine if any unexpected conditions have arisen. Information on how this interface actually works is described in later sections.

5.3 Influence of the AMRF

Much of the design described above is based on the cell controllers built in the AMRF. The first cell controller was built in 1983 [JON84]. Figure 10 shows the internal structure and external command/feedback interfaces. At that time there were only three workstations. In addition to

these workstations, the cell also interfaced with the Data Administration system (predecessor of IMDAS) and the network communications system. The cell performed three major functions: Queue Configuration Manager, Scheduling, Dispatching. The QCM assigned incoming jobs from the operator terminal to each workstation and provided feedback to the operator on the progress of each job. Since each job could be processed completely at one workstation, the QCM essentially formed three independent queues of tasks. There were one scheduler and one dispatcher for each workstation. Each scheduler sequenced the tasks in its own queue using simple rules such as First In First Out, Earliest Due Date, Shortest processing times, etc. Each dispatcher issued the next task in the sequence to its assigned workstation and monitored the feedback from that workstation. Feedback included information on the status of the workstation and the task it was performing. Tasks were dispatched one at a time and only after the preceding one was completed.

Each of the modules in the original cell controller was implemented using state tables. Figure 11 shows the simple state table which was used in one of the robot controllers. These provided a easy way to implement completely deterministic decision logic. Based on the current internal state, supervisory command, and feedback information, the table would specify the next internal state, command to all subordinates, and feedback to the supervisor. The major drawback is that they get large and complicated very quickly.

The second version of the cell [MCL87] controlled six workstations (implemented in 1986) with each job going to two more workstations. Its new design was based on the ideas in [JON85]. It differed from the original design in two ways. First, to account for the fact that a job could now require tasks to be done at more than one workstation, the function of the QCM was expanded to 1) retrieve a process plan from IMDAS for each assigned job and 2) parse that plan to determined the ordered sequence of workstations to be used in completing that job. Second, there was only one scheduler for the entire cell. It generated a queue of prioritized tasks which each dispatcher issued (again one-at-a-time) to its assigned workstation. This new version also used state tables, but not to the extent they appeared in the earlier version. In addition, it used spread sheet technology to display results on a Personal Computer.

5.4 **Impact of Material Handling**

Material Handling Systems (MHS) have a significant impact on the dynamics of a manufacturing cell. They are the primary source of coupling and can propagate delays if they are either overloaded or down due to failures. They can also play a major role in dissipating delays if other equipment is down. Hence, they are a major issue in cell controller design.

In most manufacturing cells, one of two types of material handling systems (MHS) is used: discrete (such as a robot or an AGV) and continuous (such as a conveyor) [TOM84]. Continuous material handling systems are more suitable for serial flow arrangements and are highly inflexible. Discrete MHSs are more flexible but are more challenging from a control perspective.

Consequently, the Planning and Scheduling functions must treat these MHSs as another finite capacity resource to plan and schedule [EGB84]. We believe that, from a cell control perspective, two separate material handling systems are desirable: one for intra-cell and one for inter-cell activities. We point out, however, that the scheduling of material handling systems in this type of distributed, integrated architecture has not been addressed. To do this, the class of scheduling problems must be expanded to consider MHSs as another resource to schedule at all levels. The cell's supervisor schedules inter-cell material transfers to cell load/unload stations. Each cell controller then schedules internal pickup and delivery times. The transporter scheduler uses these times to decide which transporter to use and the path, if applicable, to be used in completing the transfer. This approach will be tried in the AMRF using the framework described in [DAV88] together with the technique illustrated in [ERS86].

We note that this approach is not used in many existing facilities. Problems arise when the cell tries to plan and schedule around a critical resource that it does not own and cannot control. This can lead to the situation where no feasible schedule can be generated. AMRF researchers have recently suggested that, in situations like this, it may be beneficial to think of the material handling system as a "service" which must be shared by all. This is the same view that led to the development of the IMDAS. Perhaps a separate architecture should be developed for the MHS??? Additional research is needed to address this question.

5.5 **Impact on Process plans**

Process plans contain the information needed to manufacture, transport, and inspect parts. In the long run, both the cell and equipment level controllers will use process plans to determine how to execute assigned jobs. This requires several changes in existing process plans. First, they must have a multi-level structure which parallels the control structure. Second, process plans must provide alternate processing sequences with precedence constraints and allow backtracking when problems occur. This information is needed by both the Scheduling and Planning functions. Third, plans at different levels must have the same internal structure (AND/OR graphs are one possibility). This simplifies the software development. Finally, since computers will be responsible for processing it, this information must be provided in a consistent, error-free, and machine-readable format.

Recall that a "job" at the cell level is made up of the tasks to be executed by the various equipment in the cell. The cell level process plan will contain a "routing" for each job assigned to the cell. That routing will be a list of the equipment in the cell which can be used to manufacture the part. This includes the precedence relations that determine the alternatives sequences in which those equipment can be used. The planning function selects the run-time production plan from those alternatives. The process plan also contains timing information used by the scheduling function. The processing time for the job is the sum of the durations of the tasks that make up that job. Hence the timing data in the cell process

plan will be the aggregation of the information in the equipment level process plans. Equipment level process plans will contain the programs needed by the equipment to machine a part, move a part, inspect a part, etc. That is, they will contain NC programs, robot programs, inspection programs, etc.

5.6 External interfaces

The cell control module interfaces with 1) its subordinates and supervisor through some type of command/feedback structure, 2) the data management system, and the communications system.

5.6.1 **Command/feedback interfaces.** The cell controller must interface with its equipment controllers and its supervisor. The interface must allow for the assignment of, execution of, and monitoring of parallel activities. In addition, it must provide for start up and shutdown of all subordinates. One such command structure contains three top level fields: ACTION-VERBS, JOBS_POINTER, and RESOURCE_POINTER. Each module will have a valid set of ACTION_VERBs which initiate functions such STARTUP, SHUTDOWN, and EXECUTE_JOBS. The JOBS_POINTER parameter is a pointer to a list of jobs in the database. Each entry in this list contains a job type flag (NEW or OLD), a job ID, a job action to be taken (EXECUTE, CANCEL), process plan_IDs, priority, and limit times. The last field in this command structure is the RESOURCE_POINTER which is a reference to another list in the database. Each entry in this list refers to a specific resource request from a subordinate. It also contains the supervisor's response value (ACKNOWLEDGED, ALLOCATED, UNAVAILABLE, COMPLETED, CLEARED, etc.), and the expected time of availability.

A possible feedback structure also contains three top level fields: OPERATIONAL_STATUS, JOBS_STATUS, and RESOURCE_REQUEST. The first field, OPERATIONAL_STATUS, indicates the current operational status of the control module. The JOBS_STATUS field is used to report the evolution of all jobs assigned to the module. RESOURCE_REQUEST reports ordinary or emergency run-time resource requests to the supervisor. Each of these three fields will be further divided into two subfields: CONDITION, and POINTER. The former consists of a simple set of ASCII responses and the latter is a pointer to a more detailed list in the database. This structure reduces the complexity involved in implementation by fixing the number of input parameters and by limiting the number of values that each of those parameters can take on.

5.6.2 **Data management interface.** The cell and its equipment level subordinates need a wide variety of data to carry out their functions. There are typically three ways to access that data: each cell gets the necessary data from its supervisor and passes it to the equipment controllers; each cell has its own database management system and passes retrieved information to the equipment controllers; each controller must interface with a global database management system. The first alternative is widely used today, particularly by PLC based cell controllers. The second alternative is frequently used by the cell controllers marketed by "system integrators". These, we believe, provide a short term solution only.

The last choice is the only viable one for the future. There are three major characteristics of future manufacturing systems which support this conclusion.

First, the manufacturing environment is likely to be a heterogeneous one with equipment and cell computers purchased from a variety of vendors. This means that it is necessary to 1) provide users with a common method of accessing data, and 2) perform whatever translation, assembly, and conversion is needed to fill user requests. Second, there will be some, possibly large, number of parts which have operations performed in more than one cell. The data needed to make these parts must be shared across those cells. This means that the data system must 1) enable asynchronous interchanges of information between cell computers anywhere in the system, and 2) allow for the replication of some information units on two or more systems and the frequent and timely updates of those units. Finally, data delivery, like material delivery, is not instantaneous. It must be included in the planning of each production job. This means that data is quickly becoming a critical resource which must be scheduled. Furthermore, the scheduling decisions made by the data manager must be coordinated with scheduling decisions made at the planning and scheduling layers of the cell.

This "separate architecture" approach has an added advantage because it allows research on the two architectures to proceed independently, provided their interrelationships are well understood and accounted for in the final designs.

5.6.3 **Data communications.** Communication requirements for the components of each cell are typically of two types: very frequent command and feedback messages, and less frequent interchanges with the data management system. The command feedback messages contain relatively small amounts of data, while data interchanges contain relatively large amounts of data, such as process plans or NC programs. We believe that the physical architecture best able to handle this situation is a backbone network integrated with a subnet for each cell. The backbone network is used for inter-cell communications, access to each cell's supervisor, and the data management system. Each subnet is used for intra-cell communications. This requires each physical subnet to be transparently connected (using routers, bridges, and gateways), so that any process could conceivably communicate with any other process anywhere in the system. In addition, we recommend that messaging and protocol standards, where available, be used in the design and implementation of every component in the system.

We note three advantages to this approach. First, subnets make it easier to meet the timing requirements for tightly-coupled, intra-cell command/feedback transfers. Second, one can select the physical medium, access mechanisms, topology, and protocols for each subnet to be tailored to the needs of the particular cell it serves. Third, the backbone network is not cluttered with intra-cell communications which can negatively effect the response time of processes accessing the data management system.

5.7 Internal cell implementation structure

Figure 12 shows one possible internal implementation structure for the cell controller. It is based on the AMRF work described in [MCL87] and [CAT88] and is best implemented on a system with a multi-tasking operating system.

The Supervisor/User interface module contains the software needed to interface with the supervisor. The supervisor can be either a supervisory controller or a human operator. This module retrieves input commands from the communications manager, parses them, and passes the appropriate information to the planning module in the production manager. It also builds the feedback messages from the information provided by that planning module and forwards those messages to the communications manager to be sent back to the supervisor.

The Subordinate interface module performs similar functions. It takes information from the production manager and builds the commands to be issued to the subordinates. It also parses the feedback from the subordinates and provides that feedback to the regulation module in the production manager. Depending on the application, there may be only one subordinate interface which handles the command/feedback messages for all subordinates or separate modules for each subordinate.

The Transition manager module includes the software needed for initialization, startup, error recovery, and shutdown. This provides the internal synchronization needed to startup and shutdown both the cell itself and all its subordinates.

The production manager module contains the software needed to carry out the functions described in the planning, scheduling, and regulation layers of the cell controller. The decisions at both the planning and scheduling layers are stochastic in nature. This happens because uncertainties arise from the aggregation of information and increase in planning horizon that takes place as one moves from the equipment level to the cell level. The decisions at the regulation layer, on the other hand, are essentially deterministic. This happens because the RF assumes that subordinates will execute assigned tasks according to the prescribed plan and current schedule. Mathematical programming, simulation and expert systems have all been proposed and used to carry out these functions. We note that the framework described in [DAV88], is attractive because 1) it combines the best features of all of these techniques, 2) it already includes negotiation and compromise analysis, and 3) it can be used at all levels in the proposed architecture.

All requests to retrieve and/or update process plans, schedules, and other data that resides in the global database must go through the database interface module. That module poses the necessary "queries" in the format expected by the data administration system. When the response comes back, this module parses the incoming message and informs the internal data handler that new data has arrived.

The data handler translates all incoming data into the internal formats needed by the production manager module and stores them in the internal database. It also revises that database as needed and performs the operations necessary to convert that data back into the form used by the database to execute its updates, consistency checks, etc.

The communications interface is responsible for sending and receiving all command/feedback messages between the cell and its subordinates and all interactions with the global database. This module is the interface with the network communication system and must implement all required protocols. It must initiate the communications when the cell controller "comes up", and terminate communications before the cell "goes down".

6. SUMMARY

In this paper, we specified an architecture for a cell controller that provides a high degree of intelligence and can be easily integrated into most hierarchical shop floor control systems. It performs planning, scheduling, and regulation. We also examined cell information, data management, and communication requirements.

REFERENCES

[ALB81a] Albus, J., **Brains, Behavior, & Robotics**, McGraw Hill, New York, NY, USA, 1981.

[ALB81b] "Theory and Practice of Hierarchical Control," Albus, J. S.,Barbera, A. J., and Nagel, R. N., **Proceedings of 23rd IEEE Computer Society International Conference**, Sept. 1981.

[AXA81] Axsater, S., "Aggregation of Product Data for Hierarchical Production Planning," **Operations Research**, Vol. 29, No. 4, 1981.

[BAR89] Barkmeyer, E., "Some Interactions of Information and Control in Integrated Automation Systems", **Advanced Information Technologies for Industrial Materials Flow**, Springer-Verlag, New York, 1989.

[BEN60] Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," **Numersche Mathematik**, Vol. 4, No. 3, 1960, pp. 238-252.

[BIT77] Bitran, G. R. and Hax, A. C., "On the Design of Hierarchical Production Planning Systems", **Decision Sciences**, Vol. 8, No. 1, 28-54, 1977.

[CAT88] Catron, B. and Thomas, B., "Generic Manufacturing Controllers", **Proceedings of the IEEE Conference on Intelligent Control**, Arlington, Virginia, August, 1988.

[DAN60] Dantzig, G. B. and Wolfe, P., "Decomposition Principles for Linear Programs," **Management Science**, Vol. 8, No. 1, 1960, pp. 101-111.

[DAV88] Davis, W. and Jones, A., "A Real-time Production Scheduler for a Stochastic Manufacturing Environment", **International Journal of Computer Integrated Manufacturing,** Vol. 1, No. 2, 101-112, 1988.

[DUF86] Duffie, N. and Piper, "Non-hierarchical Control of a Flexible Manufacturing Cell", Proceedings of the International Conference on Intelligent Manufacturing Systems, Budapest, 1986.

[EGB84] Egblue, P. J. & Tanchoko, J. M. A. "Characterization of Automated Guided Vehicles Dispatching Rules", **International Journal of Production Research,** Vol. 22, No. 3, 1984.

[ERS86] Erschler, J., Roubellat, F., and Thomas, V., "Real-Time Production Scheduling for Parts with Limit Times," Technical Report No. 86063, Laboratory for Analysis of Automated Systems, Toulouse, France, 1986.

[GEF70] Geoffrion, A., "Elements of Large-Scale Mathematical Programming, Parts I and II", **Management Science,** Vol. 16, No. 11, pp. 652 - 691, 1970.

[GER89] Gershwin, S. B., "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing systems," **IEEE Proceedings Special Issue on Discrete Event Systems**, to appear 1989.

[HAT85] Hatvany, J., "Intelligence and Cooperation in Heterarchical Manufacturing Systems", **Robotics and Computer integrated Manufacturing**, Vol. 2, No. 2, 101-104, 1985

[HAX75] Hax, A. C. and Meal, A. C., "Hierarchical Integration of Production Planning and Scheduling," **Studies in Management Science,** (M. A. Geisler ed), Vol. 1, Logistics, North-Holland, 1975.

[JAM83] Jamshidi, M., **Large-Scale Systems: Modeling and Control**, North-Holland, 1983.

[JON84] Jones, A., and McLean, C., "A Cell Control System for the AMRF", **Proceedings of the International ASME Conference on Computers in Engineering**, Las Vegas, Nevada, August, 1984.

[JON85] Jones, A., and Mclean, C., "A Production Control Model for the AMRF", **Proceedings of the International ASME Conference on Computers in Engineering**, Boston, MA, August, 1985.

[JON86] Jones, A., and McLean, C. 1986, " A Proposed Hierarchical Control Model for Automated Manufacturing Systems," **Journal of Manufacturing Systems**, Vol. 5, No. 1, pp. 15-25.

[JON89] Jones, A., Barkmeyer, E., and Davis, W., 1989, "Issues in the Design and Implementation of a System Architecture for Computer Integrated Manufacturing", **International Journal of Computer Integrated Manufacturing special issue on CIM architecture,** 2(2), 65-76.

[JON90] Jones, A. and Saleh A., "A Multi-level/Multi-layer Architecture

for Intelligent Shop Floor Control", **International Journal of Computer Integrated Manufacturing special issue on Intelligent Control,** (to appear)

[LIB88] Libes, D. and Barkmeyer, E., " The Integrated Manufacturing Data Administration System (IMDAS) - An Overview", **International Journal of Computer Integrated Manufacturing,** Vol. 1, No. 1, pp. 44-49, 1988.

[MCL87] McLean, C. R., "A Cell Control Architecture for Flexible Manufacturing", **Proceedings of the Advanced Manufacturing Systems Conference,** Chicago, Illinois, June, 1987.

[MES70] Mesarovic, M. D., Macleo, D., and Takahara, Y., **Theory of Multilevel Hierarchical Systems,** New York, Academic Press, 1970.

[RYB88] Rybczynski, S., Wallace, E., Libes, D., Barkmeyer, E., Strawbridge, M., and Young, C., "AMRF Network Communications", NBSIR 88-3816, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[SAN78] Sandell, N., Variya, R., Athans, M., and Safanov, M., "A Survey of Decentralized Control Methods for Large Scale Systems," **IEEE Transactions on Automatic Control,** Vol. AC-23, No. 2, 1978.

[SAR85] Saridis, G., "Foundations of the Theory of Intelligent Controls", IEEE Workshop on Intelligent Control, Troy, New York, August, 1985.

[SIM82] Simpson, J., Hocken, R., and Albus J., "The Automated Manufacturing Research Facility of the National Bureau of Standards", **Journal of Manufacturing Systems,** Vol. 1, No. 1, 1982.

[TOM84] Tompkins, J. A. & White, J. A., Facilities Planning, John Wiley, 1984.

[VIL86] Villa, A. and Rossetta, S., "Towards a Hierarchical Structure for Production Planning and Control in Flexible Manufacturing Systems", **Modeling and Design of Flexible Manufacturing Systems** (A. Kusiak ed), Elsevier Science, Amsterdam, 1986.

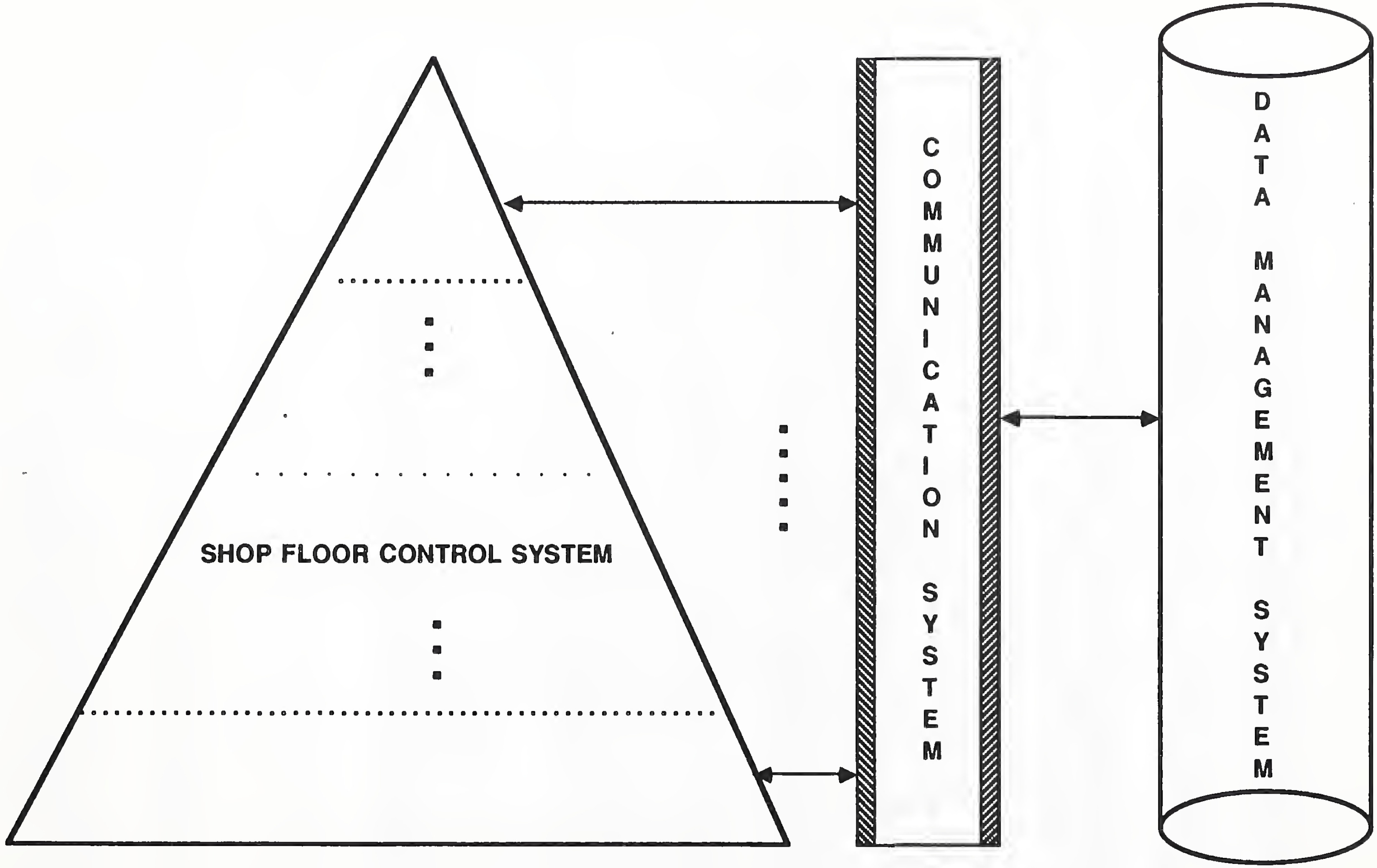

FIGURE 1 SEPARATE CONTROL, COMMUNICATION, DATA MANAGEMENT SYSTEMS

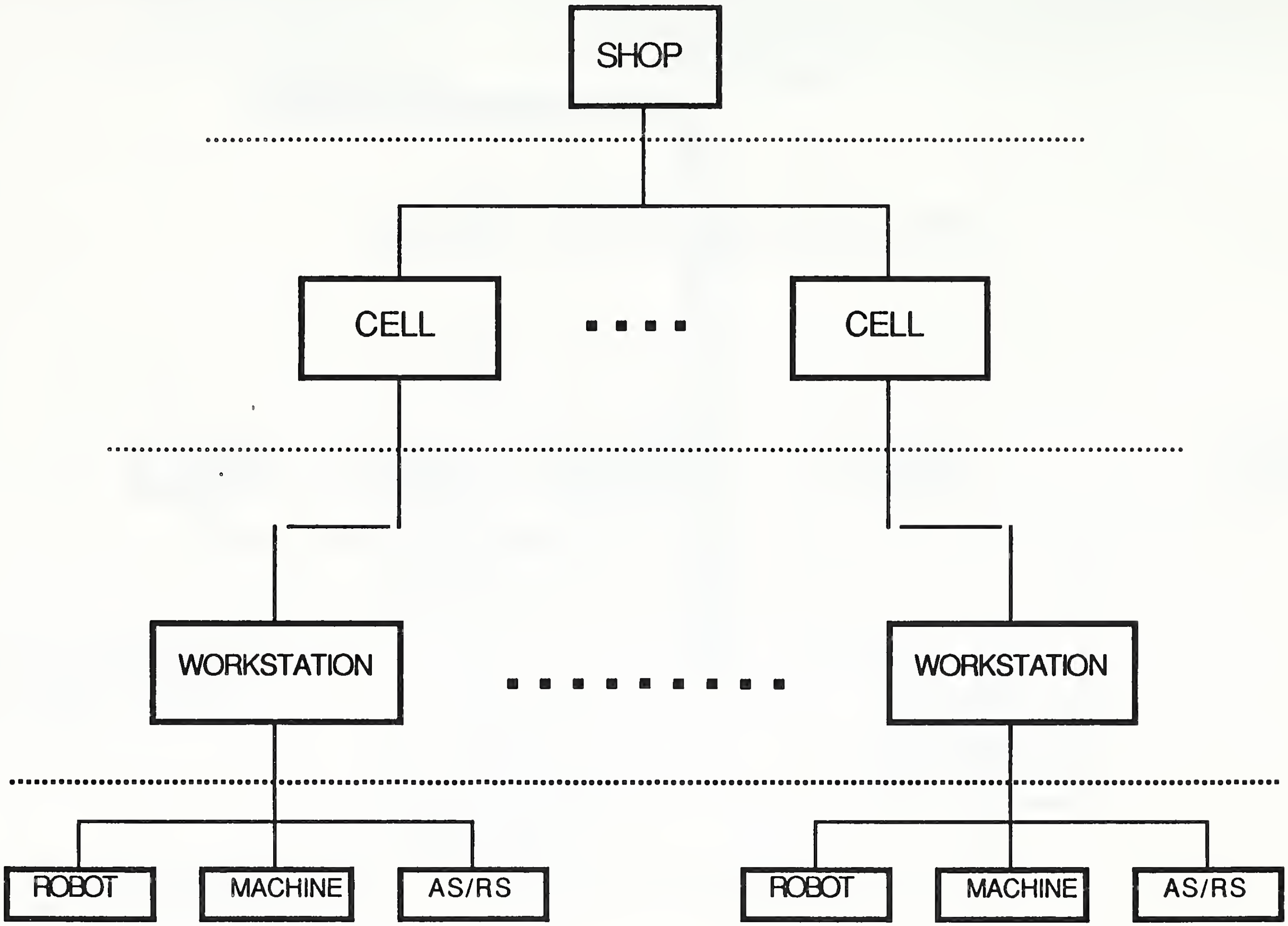

FIGURE 2 AMRF SHOP FLOOR CONTROL ARCHITECTURE

MDAS - INTER-DDAS QUERY MNANAGEMENT AND
GLOBAL DATA DISTRIBUTION DICTIONARY

DDAS - INTER BDAS QUERY MANAGEMENT,
DATA DICTIONARY, USER INTERFACE

BDAS - INTERFACE TO DBMs, DATA
TRANSLATION, AND DATA
DELIVERY TO OTHER BDASs

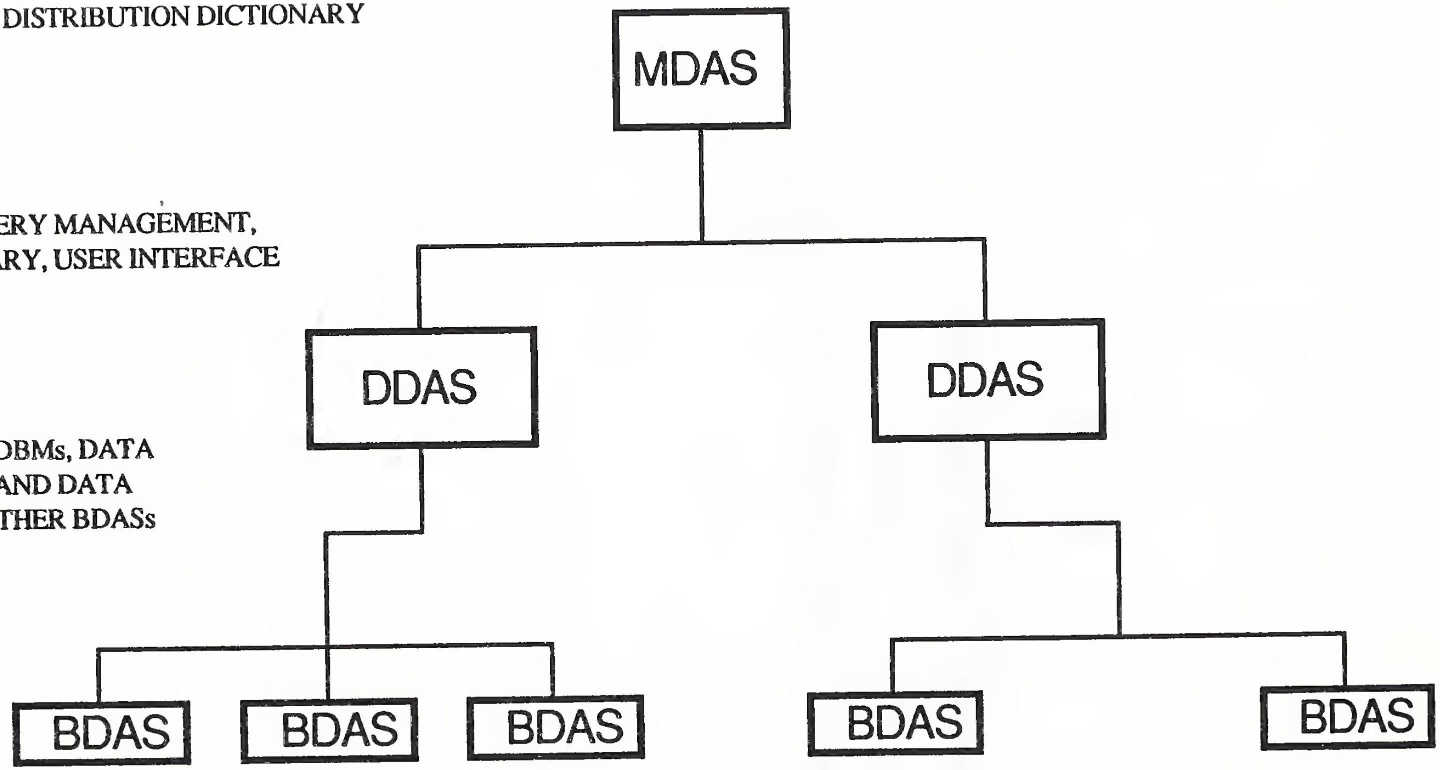

FIGURE 3 IMDAS ARCHITECTURE

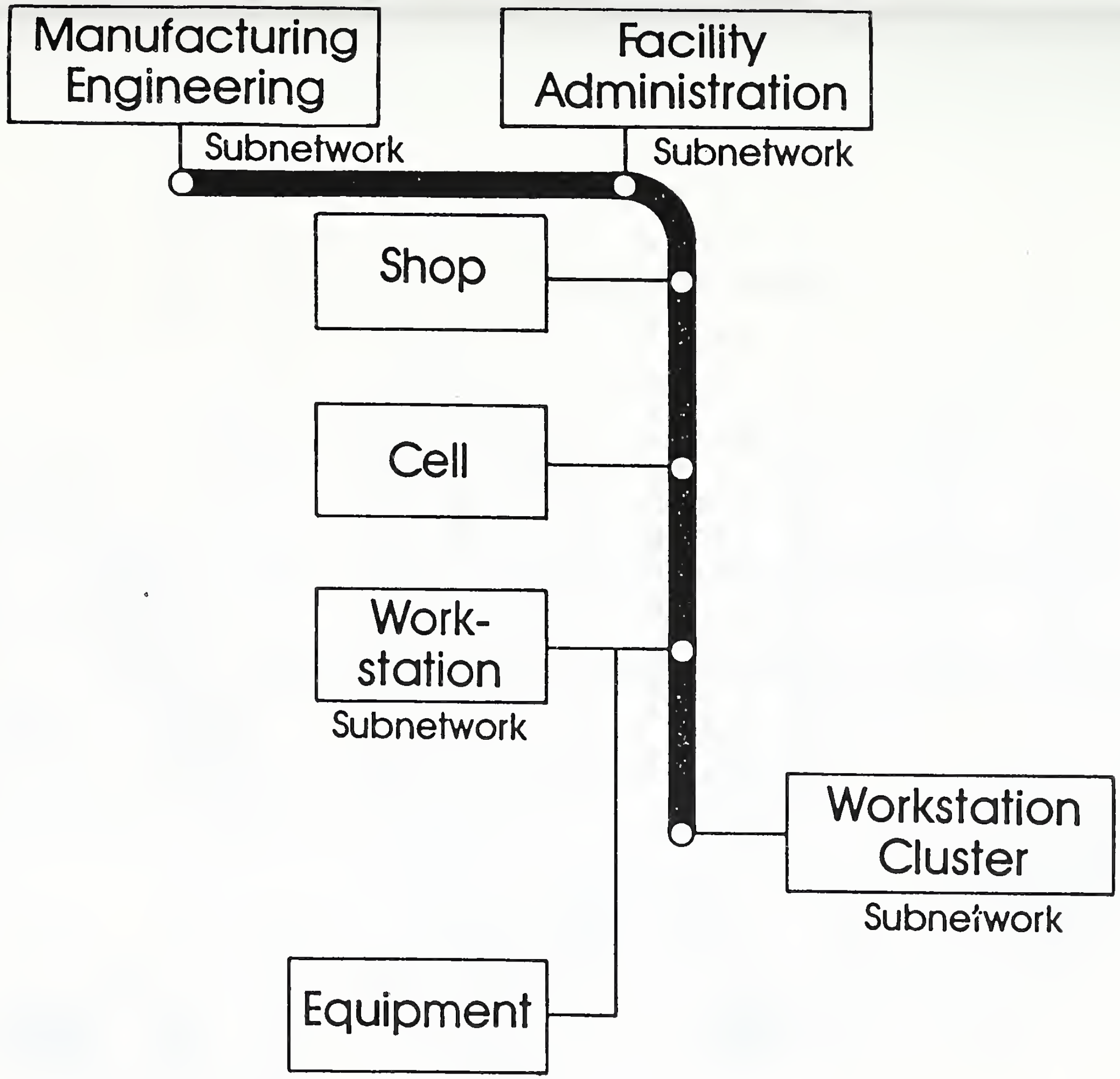

FIGURE 4 AMRF NETWORK ARCHITECTURE

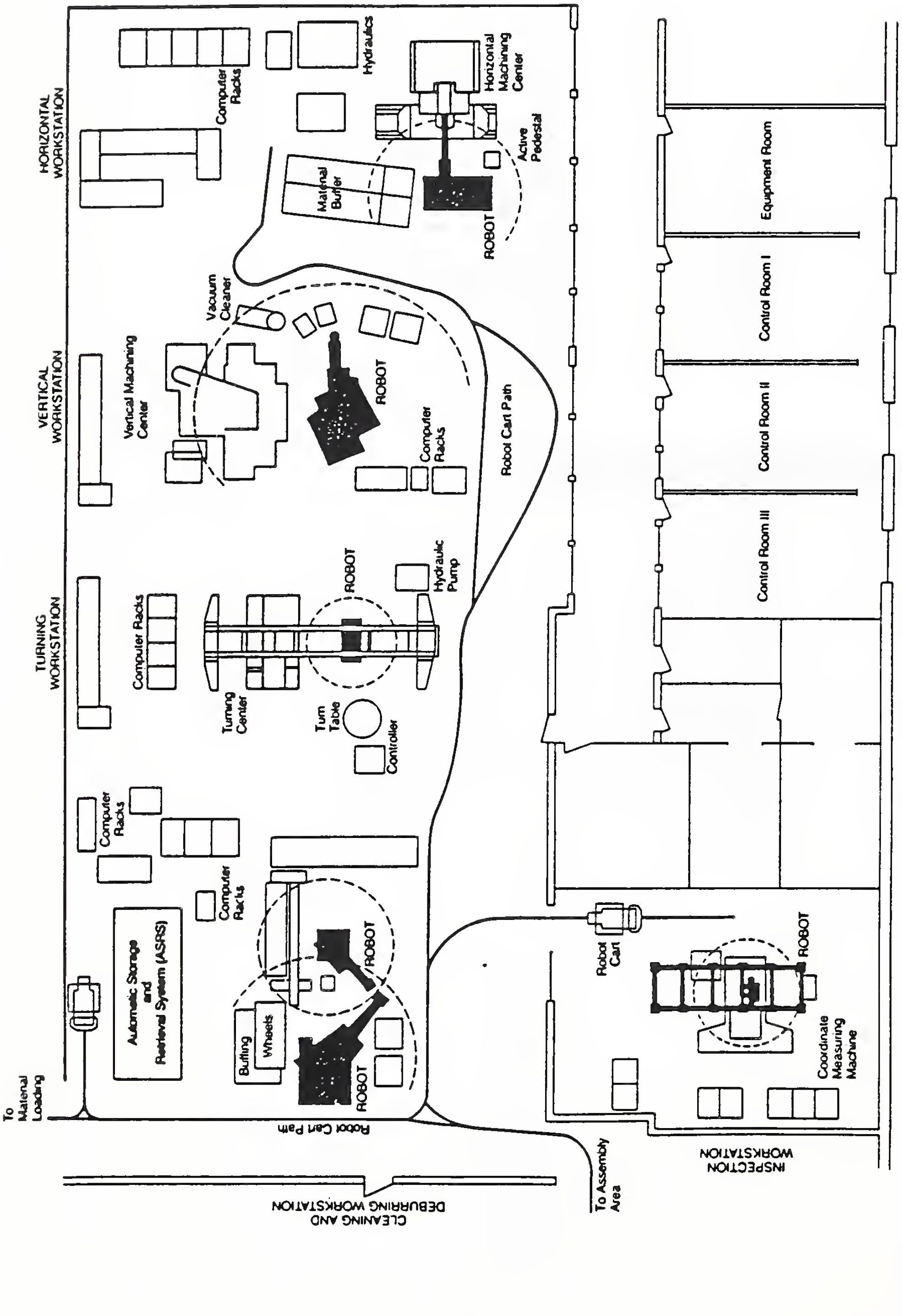

FIGURE 5 AMRF SHOP FLOOR LAYOUT

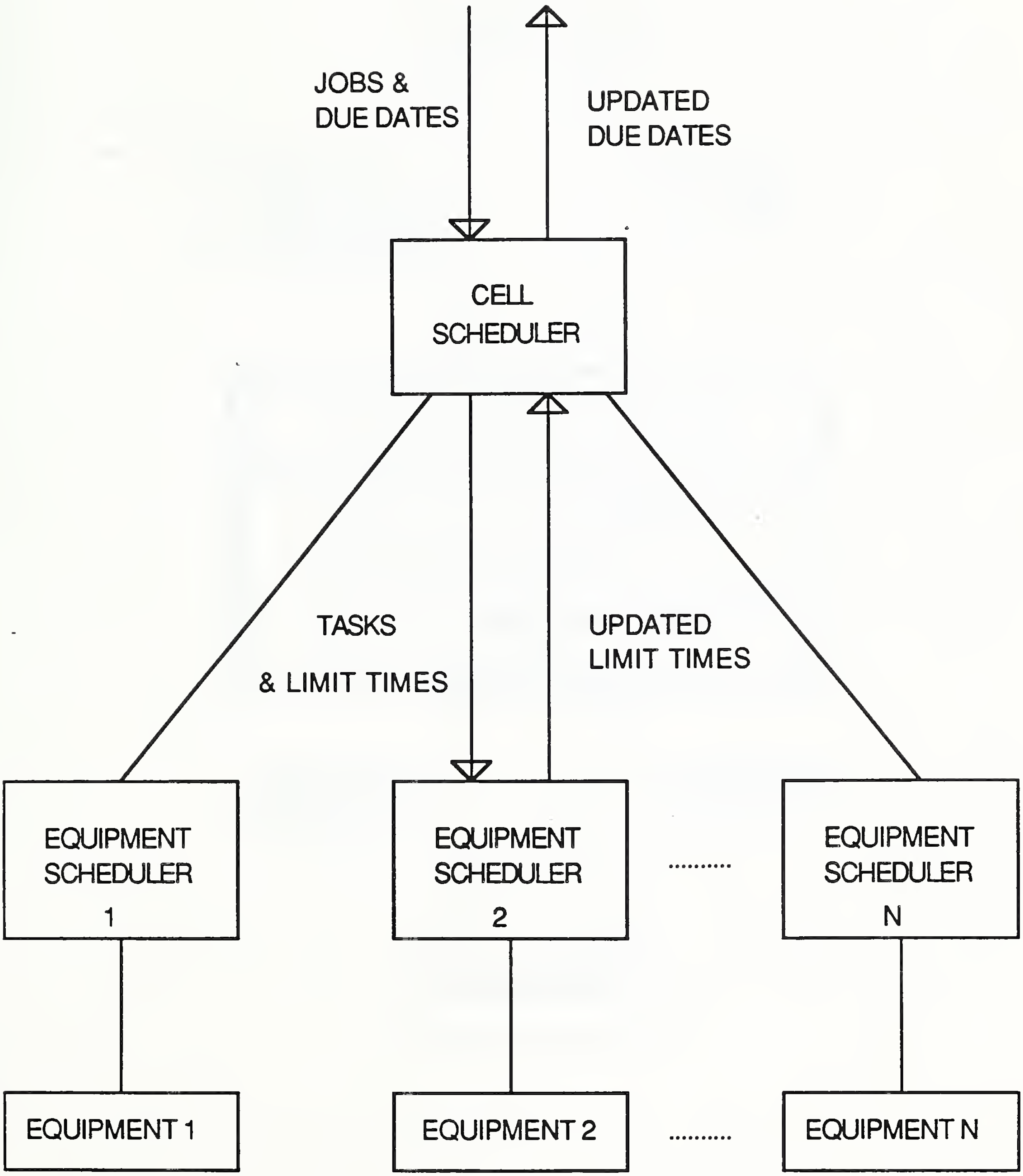

FIGURE 6 DAVIS & JONES TWO LEVEL SCHEDULER

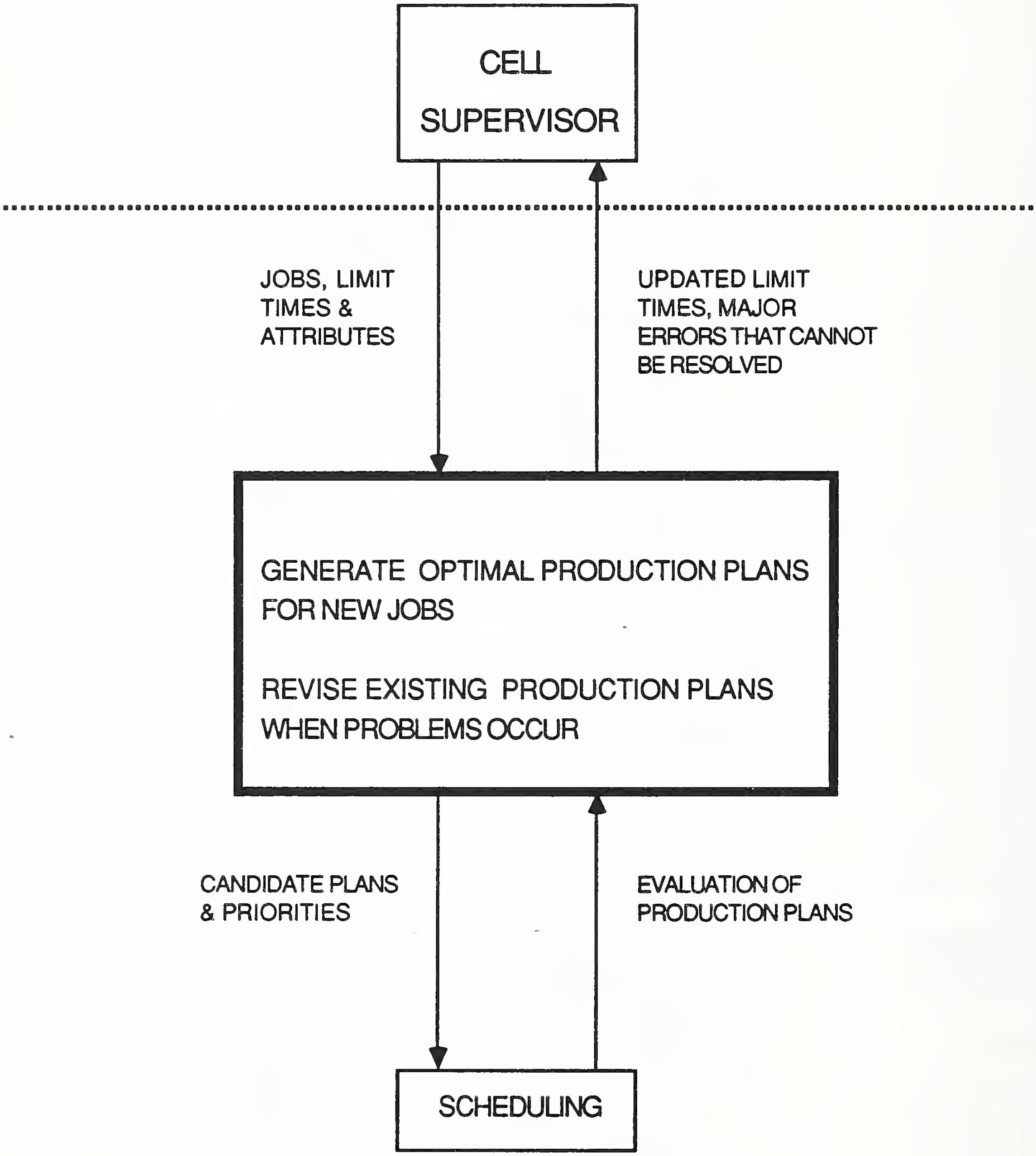

FIGURE 7 PLANNING FUNCTION

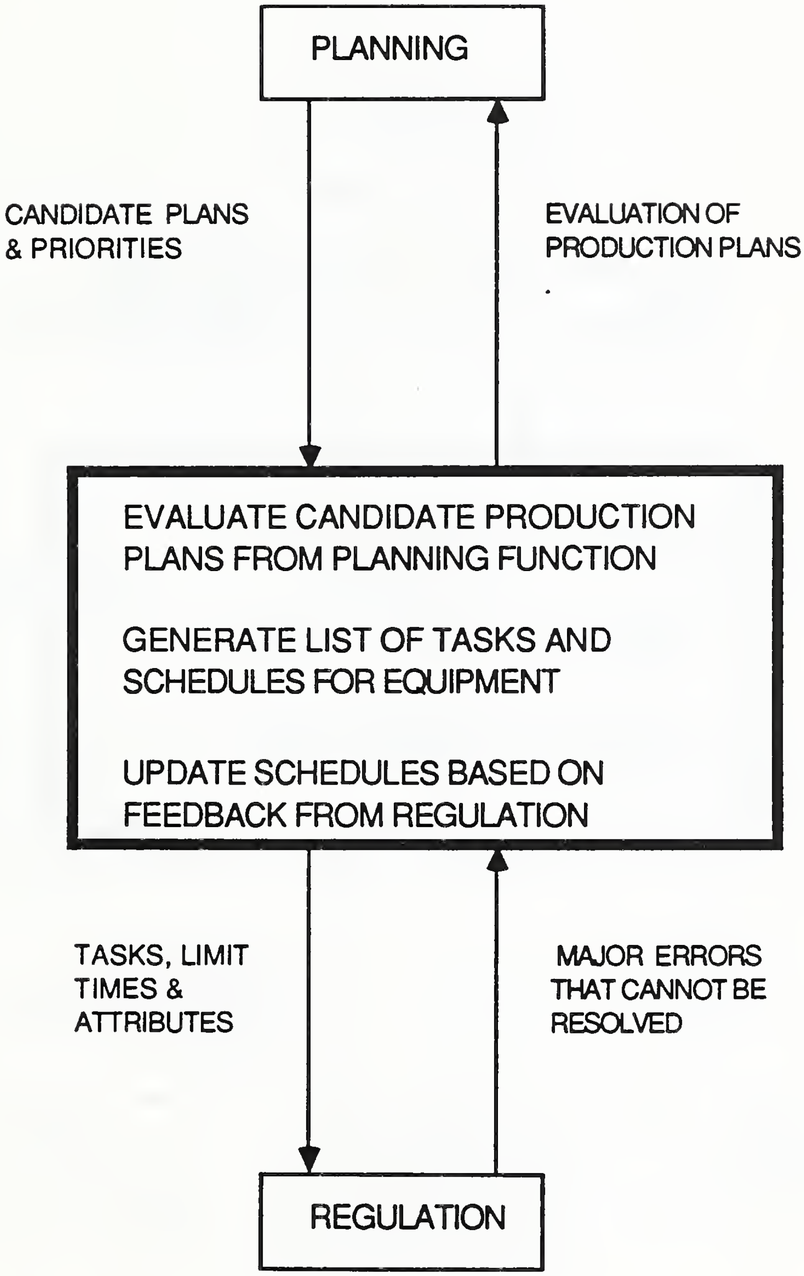

FIGURE 8 SCHEDULING FUNCTION

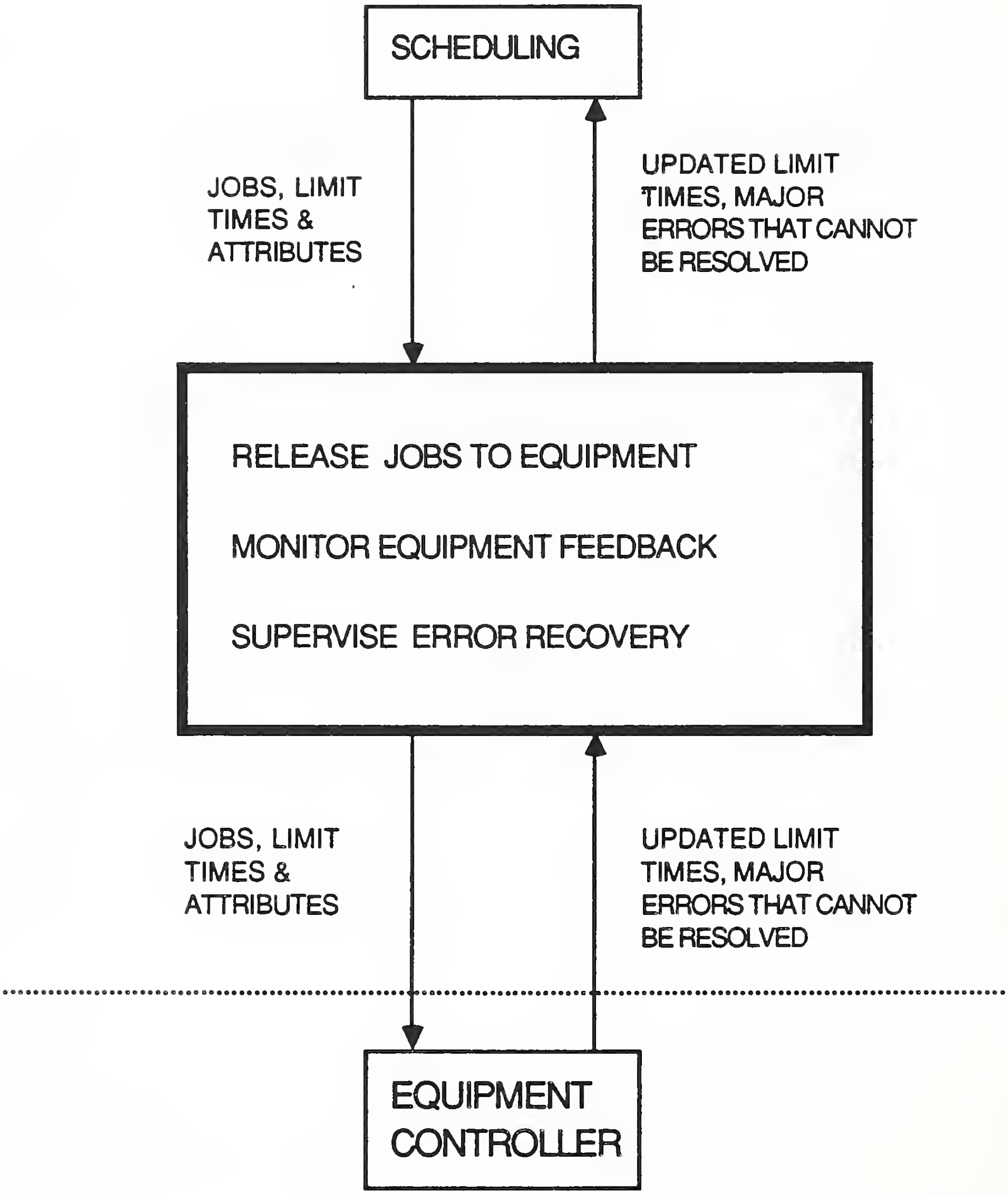

FIGURE 9 REGULATION FUNCTION

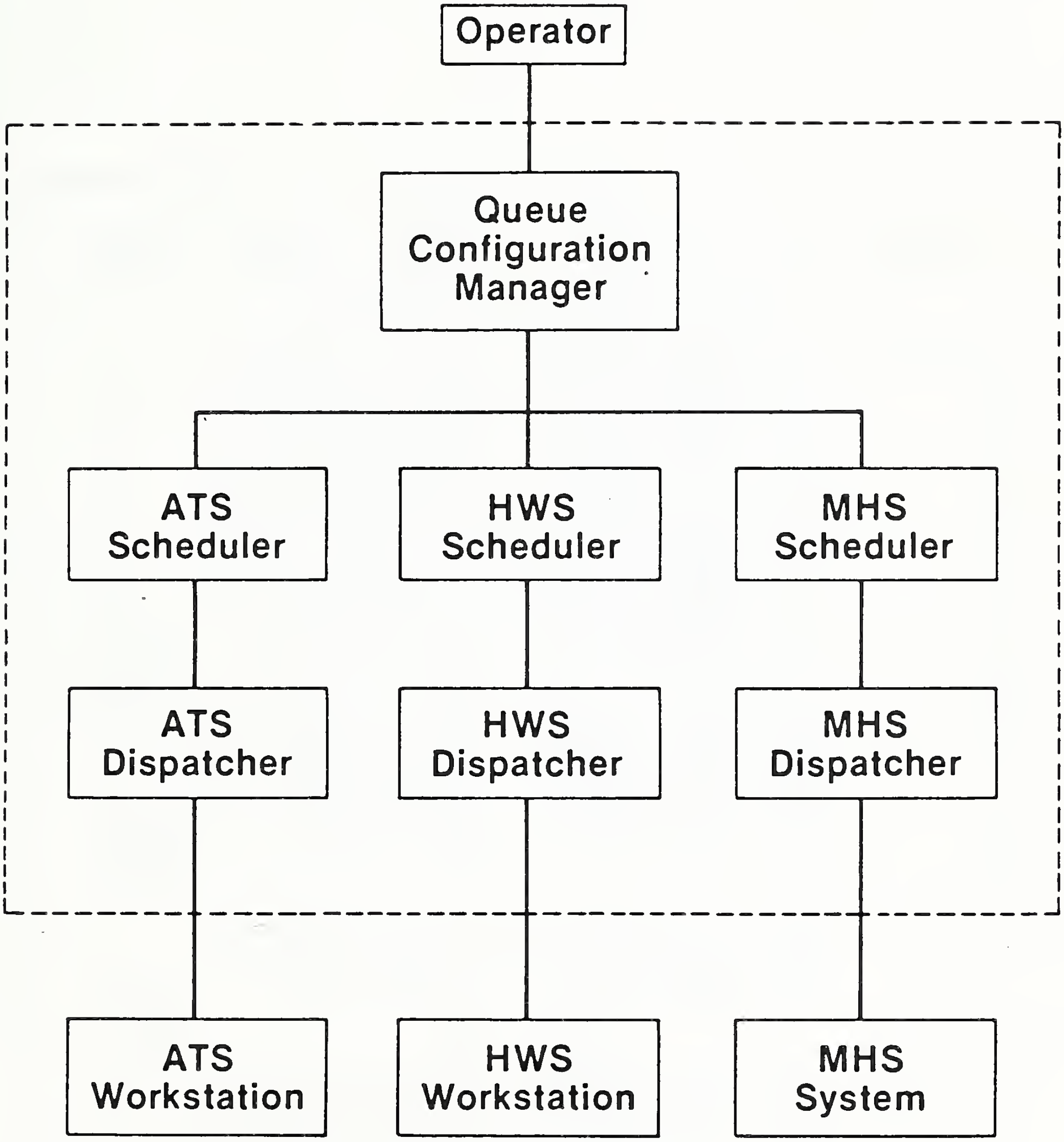

FIGURE 10 FIRST AMRF CELL CONTROLLER

Command	State	Feedback	Next State	Output	Report
—	C30	No New Command	C30	Wait	—
Fetch (A)	C30	New Command	C31	Reach to (A)	—
"	C31	Distance to $A > T1$	C31	Reach to (A)	—
"	C31	Distance to $A \leqslant T1$	C32	Grasp (A)	—
"	C31	A Not Visable	C35	Search for (A)	—
"	C32	Grasp Pressure $< T2$ Grip Dist $> T3$	C32	Grasp (A)	—
"	C32	Grasp Pressure $\geqslant T2$ Grip Dist $> T3$	C33	Move to (X)	—
"	C32	Grip Dist $\leqslant T3$	C36	Back Up (Y)	Object Missing
"	C33	Distance to $X > 0$	C33	Move to (X)	—
"	C33	Distance to $X = 0$	C34	Release	—
"	C34	Grip Dist $< T4$	C34	Release	—
"	C34	Grip Dist $\geqslant T4$	C30	Wait	Report Fetch Done
"	C35	A Not Visable	C35	Search for (A)	—
"	C35	A in Sight	C31	Reach to (A)	—
"	C35	Search Fail	C30	Wait	Report Fetch Fail
"	C36	Back Up Not Done	C36	Back Up (Y)	—
"	C36	Back Up Done	C35	Search for (A)	—

FIGURE 11 STATE TABLE TO IMPLEMENT THE ROBOT COMMAND "FETCH (A)"

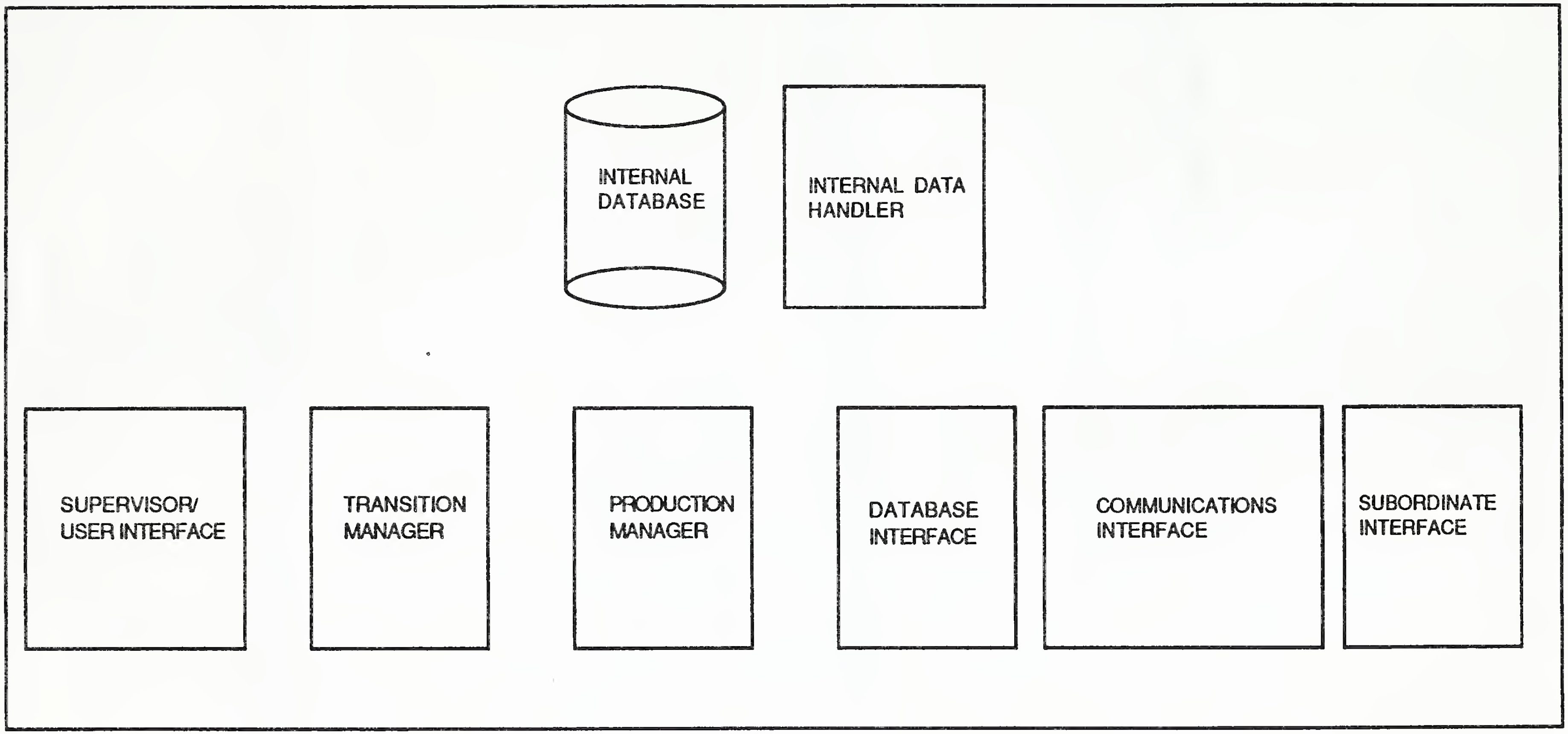

FIGURE 12 INTERNAL CELL STRUCTURE

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4308

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
MAT MAY 1990

4. TITLE AND SUBTITLE
THE CELL AS PART OF A MANUFACTURING SYSTEM

5. AUTHOR(S)
Albert Jones and Abdol Saleh

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)
NIST, Gaithersburg, MD 20899
AT+T Laboratories

10. SUPPLEMENTARY NOTES

☐ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

In this chapter, we describe a new approach to the design, implementation, and integration of cell controllers in a manufacturing system. It combines techniques from control theory, operations research, and computer science. This cell controller can be 1) modified to fit both the physical definition of a cell and the capabilities of other controllers in the system, and 2) easily integrated into any shop floor control system which meets the interface requirements.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)
Automation; cell control; control theory; planning; scheduling; integration.

13. AVAILABILITY
- X UNLIMITED
- ☐ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
- ☐ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
- X ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
32

15. PRICE
A03

ELECTRONIC FORM